Part 1: Выбор Сценария

Для данной работы выбран сценарий: Сервис потоковой передачи музыки: Управление артистами, альбомами, песнями, пользователями и пользовательскими плейлистами.

Part 2: Проектирование Базы Данных и Документация

**Идентификация Сущностей и Атрибутов:**

1. **Исполнитель (Singer):**
2. **Альбом (Album):**
3. **Песня (Song):**
4. **Жанр (Genre):**
5. **Пользователь (User)**
6. **Плейлист (Playlist)**
7. **Подписка (Subscription)**

**Проектирование Таблиц:**

1. **Table Name: Singer**

**Description:** Хранит информацию об исполнителях

**Attributes:**

* *SingerID*: INTEGER, PK, NOT NULL, UNIQUE
* *Name*: VARCHAR(50), NOT NULL, UNIQUE
* *BIO*: VARCHAR(1000), NOT NULL
* *GenreID*: INTEGER, FK (REFERENCES Genre), NOT NULL

**Constraints:**

* *PK\_Singer*: PRIMARY KEY (SingerID)
* *FK\_Singer\_Ganre*: FOREIGN KEY (GenreID) REFERENCES Genre (GenreID)
* *UQ\_SingerName*: UNIQUE (Name)

1. **Table Name: Album**

**Description**: Хранит информацию об альбомах

**Attributes**:

* *AlbumID*: INTEGER, PK, NOT NULL, UNIQUE
* *Album Name*: VARCHAR(30), NOT NULL
* *ReleaseDate*: DATE, NOT NULL
* *SingerID*: INTEGER, FK (REFERENCES Artist), NOT NULL
* *CoverUrl*:TEXT, NOT NULL, UNIQUE

**Constraints:**

* *PK\_Album*: PRIMARY KEY (AlbumID)
* *FK\_Album\_Singer*: FOREIGN KEY (SingerID) REFERENCES Singer (SingerID)
* *UQ\_CoverUrl*: UNIQUE (CoverUrl)

1. **Table Name: Genre**

**Description:** Хранит информацию о жанрах

**Attributes:**

* *GenreID:* INTEGER, PK, NOT NULL, UNIQUE
* *GenreName*: VARCHAR(50), NOT NULL, UNIQUE

**Constraints:**

* *PK\_Genre*: PRIMARY KEY (GenreID)
* *UQ\_GanreName*: UNIQUE (GenreName)

1. **Table Name: Song**

**Description:** Хранит информацию о песнях

**Attributes:**

* *SongID*: INTEGER, PK, NOT NULL, UNIQUE
* *SongName*: VARCHAR(50), NOT NULL
* *Duration*: INTEGER, NOT NULL, CHECK (duration > 0)
* *SingerID*: INTEGER, FK (REFERENCES Singer), NOT NULL
* *AlbumID*: INTEGER, FK (REFERENCES Album), NOT NULL
* *AudioUrl*: TEXT, NOT NULL, UNIQUE
* *GenreID*: INTEGER, FK (REFERENCES Genre), NOT NULL

**Constraints:**

* *PK\_Song*: PRIMARY KEY (SongID)
* *CHK\_DurationSong*: CHECK (duration > 0)
* *FK\_Song\_Singer*: FOREIGN KEY (SingerID) REFERENCES Singer (SingerID)
* *FK\_Song\_Album*: FOREIGN KEY (AlbumID) REFERENCES Album (AlbumID)
* *FK\_Song\_Genre*: FOREIGN KEY (GenreID) REFERENCES Genre (GenreID)
* *UQ\_AudioUrl*: UNIQUE (AudioUrl)

1. **Table Name: User**

**Description:** Хранит информацию о пользователях

**Attributes:**

* *UserID*: INTEGER, PK, NOT NULL, UNIQUE
* *UserName*: VARCHAR(50), NOT NULL, UNIQUE
* *Email*: VARCHAR(50), NOT NULL, UNIQUE
* *Password*: INTEGER, NOT NULL
* *SubscriptionID*: INTEGER, FK (REFERENCES Subscription), NOT NULL

**Constraints:**

* *PK\_User*: PRIMARY KEY (UserID)
* *FK\_Subscription*: FOREIGN KEY (SubscriptionID) REFERENCES Subscription (SubscriptionID)
* *UQ\_ UserName*: UNIQUE (UserName)
* *UQ\_Email*: UNIQUE (Email)

1. **Table Name: Playlist**

**Description:** Хранит информацию о плейлистах

**Attributes:**

* *PlaylistID*: INTEGER, PK, NOT NULL, UNIQUE
* *Name*: VARCHAR(50), NOT NULL
* *UserID*: INTEGER, FK (REFERENCES User), NOT NULL
* *IsPublic*: BOOLEAN, NOT NULL DEFAULT FALSE

**Constraints:**

* *PK\_Playlist*: PRIMARY KEY (PlaylistID)
* *FK\_User:* INTEGER, FOREING KEY (UserID) REFERENCES User (UserID)

1. **Table Name: Subscription**

**Description:** Хранит информацию о типе подписки

**Attributes:**

* *SubscriptionID*: INTEGER, PK, NOT NULL, UNIQUE
* *Type*: VARCHAR(30), NOT NULL, UNIQUE

**Constraints:**

* *PK\_Subscription*: PRIMARY KEY (SubscriptionID)
* *UQ\_SubscriptionType*: UNIQUE (Type)

**Взаимосвязи:**

* **Singer и Album (Один-ко-многим):** Один исполнитель может выпустить несколько альбомов, но каждый альбом в упрощенном виде может иметь одного исполнителя.
  + Album.SingerID является внешним ключом, ссылающимся на Singer.SingerID
* **Singer и Song (Один-ко-многим):** Один артист может записать множество песен, в упрощенном варианте множество песен могут иметь одного исполнителя.
* Song.SingerID является внешним ключом ссылающимся на Singer.SingerID
* **Album и Song (Один-ко-многим):** Один альбом может содержать множество песен, множество песен могут находиться в одном альбоме.
  + Song.AlbumID является внешним ключом ссылающимся на Album.AlbumID
* **Genre Singer/Song (Один-ко-многим):** Один жанр может быть связан с множеством артистов и песен, множество артистов и песен могут быть связаны с одним жанром.
  + Song.GenreID является внешним ключом ссылающимся на Genre.GenreID
  + Singer.GenreID является внешним ключом ссылающимся на Genre.GenreID
* **User и Playlist (Один-ко-многим):** Один пользователь может иметь множество плейлистов, в упрощенном варианте множество плейлистов могут принадлежать одному пользователю
  + Playlist.UserID является внешним ключом ссылающимся на User.UserID
* **User и Subscription (Многие-к-одному):** Многие пользователи могут иметь один и тот же тип подписки
  + User.SubscriptionID является внешним ключом ссылающимся на Subscription.SubscriptionID
* **Playlist и Song (Многие-ко-многим):** Один плейлист может содержать множество песен, одна песня может содержаться в нескольких плейлистах.

Через промежуточную таблицу:

* + Playlist\_Songs.PlaylistID является внешним ключом, ссылающимся на Playlist.PlaylistID
  + Playlist\_Songs.SongID является внешним ключом, ссылающимся на Song.SongID

Part 3: ER-Диаграмма

